

# **The RSA Validation System (RSAVS)**

April 20, 2004

Sharon S. Keller

National Institute of Standards and Technology  
Information Technology Laboratory  
Computer Security Division



## TABLE OF CONTENTS

<u>1</u>	<u>INTRODUCTION</u>	2
<u>2</u>	<u>SCOPE</u>	2
<u>3</u>	<u>CONFORMANCE</u>	3
<u>4</u>	<u>DEFINITIONS AND ABBREVIATIONS</u>	3
<u>4.1</u>	<u>DEFINITIONS</u>	3
<u>4.2</u>	<u>ABBREVIATIONS</u>	3
<u>5</u>	<u>DESIGN PHILOSOPHY OF THE RSA VALIDATION SYSTEM</u>	4
<u>6</u>	<u>RSAVS TESTS</u>	4
<u>6.1</u>	<u>CONFIGURATION INFORMATION</u>	4
<u>6.2</u>	<u>SIGNATURE GENERATION TEST</u>	5
<u>6.6</u>	<u>SIGNATURE VERIFICATION TEST</u>	6
<u>APPENDIX A</u>	<u>REFERENCES</u>	8
<u>APPENDIX B</u>	<u>EXAMPLE OF REQUEST, FAX, RESPONSE, AND SAMPLE FILES FOR RSA AS APPROVED IN FIPS186-2 AND SPECIFIED IN ANSI X9.31</u>	9
<u>B.1</u>	<u>Examples of REQUEST Files</u>	9
<u>B.1.1</u>	<u>SigGenRSA.req</u>	9
<u>B.1.2</u>	<u>SigVerRSA.req</u>	10
<u>B.2</u>	<u>EXAMPLES OF FAX FILES</u>	13
<u>B.2.1</u>	<u>SigGenRSA.fax</u>	13
<u>B.2.2</u>	<u>SigVerRSA.fax</u>	14
<u>B.3</u>	<u>EXAMPLES OF RESPONSE FILES</u>	18
<u>B.3.1</u>	<u>SigGenRSA.rsp</u>	18
<u>B.3.2</u>	<u>SigVerRSA.rsp</u>	21
<u>B.4</u>	<u>EXAMPLES OF SAMPLE FILES</u>	23
<u>B.4.1</u>	<u>SigGenRSA.sam</u>	23
<u>B.4.2</u>	<u>SigVerRSA.sam</u>	25

## 1 Introduction

This document, *The RSA Validation System (RSAVS)* specifies the procedures involved in validating implementations of public key cryptography based on the RSA algorithm. This document deals with three variations of the RSA algorithm. They are:

- RSA algorithm specified in FIPS186-2, *Digital Signature Standard (DSS)*, January 27, 2000[1], and
- Two signature schemes with appendix specified in *Public Key Cryptography Standards(PKCS) #1 v2.1: RSA Cryptography Standard-2002* [2]. These two signature schemes with appendix are
  - RSASSA-PSS, and
  - RSASSA-PKCS1-v1\_5.

The RSAVS only supports the RSA algorithm; that is, it only supports implementations where the public verification exponent  $e$  is odd. Both X9.31 and the PKCS #1 V2.1 documents support RSA. As specified in ANSI X9.31-1998, Section 4.1.1 (as pointed to by FIPS186-2), when the public key exponent is odd, the digital signature algorithm is commonly called RSA. When the public key exponent is even, the digital signature algorithm is commonly called Rabin-Williams.

The RSAVS is designed to perform automated testing on Implementations Under Test (IUTs). This document provides the basic design and configuration of the RSAVS. Included are the specifications for testing the Signature Generation and Signature Verification components of the IUT.

This document defines the purpose, the design philosophy, and the high-level description of the validation process for RSA. The requirements and administrative procedures to be followed by those seeking formal validation of an implementation of RSA are presented. The requirements described include the specification of the data communicated between the IUT and the RSAVS, the details of the tests that the IUT must pass for formal validation, and general instruction for interfacing with the RSAVS. Additionally, an appendix is also provided containing samples of input and output files for the RSAVS.

## 2 Scope

This document specifies the tests required to validate IUTs for conformance to the RSA as specified in [1] and [2]. When applied to IUTs that implement any of the three different algorithm variations of the RSA, the RSAVS provides testing to determine the correctness of the signature generation and verification components contained in the implementation. These two separate tests examine the signature generation and the signature verification algorithm

components. In addition to determining conformance to the cryptographic specifications, the RSAVS is structured to detect implementation flaws including pointer problems, insufficient allocation of space, improper error handling, and incorrect behavior of the RSA implementation.

### **3 Conformance**

The successful completion of the tests contained within the RSAVS is required to be validated as conforming to the RSA. Testing for the cryptographic module in which the RSA is implemented is defined in FIPS PUB 140-2, *Security Requirements for Cryptographic Modules*.[3]

### **4 Definitions and Abbreviations**

#### **4.1 Definitions**

<b>DEFINITION</b>	<b>MEANING</b>
CMT laboratory	Cryptographic Module Testing laboratory that operates the RSAVS
RSA algorithm	The algorithm specified in the FIPS186-2, <i>Digital Signature Standard (DSS)</i> and the PKCS#1 v2.1 document.
PKCS	Public Key Cryptography Standards

#### **4.2 Abbreviations**

<b>ABBREVIATION</b>	<b>MEANING</b>
RSA	RSA algorithm specified in FIPS186-2
RSAVS	RSA Validation System
IUT	Implementation Under Test
PKCS	Public Key Cryptography Standards
RSASSA	RSA Signature Scheme with appendix
RSASSA-PKCS1_V1_5	PKCS # 1 Version 1.5 Signature Scheme with appendix
RSASSA-PSS	Probabilistic Signature Scheme with appendix

## **5 Design Philosophy Of The RSA Validation System**

The RSAVS is designed to test conformance to RSA rather than provide a measure of a product's security. The validation tests are designed to assist in the detection of accidental implementation errors, and are not designed to detect intentional attempts to misrepresent conformance. Thus, validation should not be interpreted as an evaluation or endorsement of overall product security.

The RSAVS has the following design philosophy:

1. The RSAVS is designed to allow the testing of an IUT at locations remote to the RSAVS. The RSAVS and the IUT communicate data via *REQUEST* and *RESPONSE* files.
2. The testing performed within the RSAVS utilizes statistical sampling (i.e., only a small number of the possible cases are tested); hence, the successful validation of a device does not imply 100% conformance with the standard.

## **6 RSAVS Tests**

The RSAVS provides conformance testing for two of the components of the algorithm, as well as testing for apparent implementation errors. The components tested are signature generation and signature validation.

### **6.1 Configuration Information**

To initiate the validation process of the RSAVS, a vendor submits an application to an accredited laboratory requesting the validation of its implementation of RSA. The vendor's implementation is referred to as the Implementation Under Test (IUT). The request for validation includes background information describing the IUT along with information needed by the RSAVS to perform the specific tests. More specifically, the request for validation includes:

1. Vendor Name;
2. Product Name;
3. Product Version;
4. Implementation in software, firmware, or hardware;
5. Processor and Operating System with which the IUT was tested if the IUT is implemented in software or firmware;

6. Brief description of the IUT or the product/product family in which the IUT is implemented by the vendor (2-3 sentences); and
7. The modulus size(s) supported by the IUT.
8. A SHA algorithm supported by the implementation.
9. For RSASSA-PSS implementations, a SALT length.

## 6.2 Signature Generation Test

An implementation of the RSA may generate the digital signature. This option tests the ability of an IUT to produce correct signatures. To test signature generation, the RSAVS supplies ten messages to the IUT. The IUT generates the corresponding signatures and returns them to the RSAVS. The RSAVS validates the signatures by using the associated public key to verify the signature.

The RSAVS:

- A. Creates a *REQUEST* file (Filename: RSASigGen.req) containing:
  1. The Product Name and
  2. Ten messages to be signed for each modulus size supported.

Note: The CMT laboratory sends the *REQUEST* file to the IUT.

The IUT:

- A. Generates the signatures for the messages supplied in the *REQUEST* file.
- B. Creates a *RESPONSE* file (Filename: RSASigGen.rsp) containing:
  1. The Product Name,
  2. The modulus,  $n$ ,
  3. The public key,  $e$ , corresponding to the private key,  $d$ , used to generate the signatures and
  4. The ten messages,  $Msg$ , and their corresponding signature values,  $s$ .

Note: The IUT sends the *RESPONSE* file to the CMT laboratory for processing by the RSAVS.

The RSAVS:

- A. Uses the respective public keys to verify the signatures in the *RESPONSE* file.
- B. Records PASS for this test if all conditions are met; otherwise, records FAIL.

## 6.6 Signature Verification Test

This option tests the ability of the IUT to recognize valid and invalid signatures. For each modulus size selected, the RSAVS generates a modulus and three associated key pairs,  $(d, e)$ . Each private key  $d$  is used to sign four pseudorandom messages each of 1024 bits. Some of the public keys,  $e$ , messages or signatures are altered so that signature verification should fail. The messages, signatures, modulus and public key  $e$  values are forwarded to the IUT. The IUT then attempts to verify the signatures and returns the results to the RSAVS, which compares the received results with its own stored results.

The RSAVS:

- A. Generates 3 groups of data for each supported modulus size. Each group consists of a modulus and 4 sets of data. Each set of data contains
  - 1. A pseudorandom message,
  - 2. A public/private key pair that is consistent with the modulus and
  - 3. A signature for the message using the private key.
- For the efficiency of the tool, a modulus will sometimes be used for more than one group of data when it is consistent with more than one public key. Therefore when loading the modulus for each group of data, the modulus specified for this group may be the same as that specified for the previous group.
- B. Alters the public key, the message or the signature for three fourths of the public key/message/signature sets such that the message verification fails.
- C. Creates a *REQUEST* file (Filename: RSASigVer.req) containing:
  - 1. The Product Name;
  - 2. 3 groups of data consisting of
    - a. The modulus  $n$  for the supported modulus size,
    - b. The information from step B, including:
      - i. A public key corresponding to the private key used to sign the messages,
      - ii. The pseudorandom message and
      - iii. The signature.

Note: The CMT laboratory sends the *REQUEST* file to the IUT.

- D. Creates a *FAX* file (Filename: RSASigVer.fax) containing:
  - 1. The information from the *REQUEST* file and

- 
2. An indication of whether the signature verification process should pass or fail, for each public key/message/signature set.

The IUT:

- A. Attempts to verify the signatures for the messages supplied in the *REQUEST* file using the corresponding modulus  $n$  and the public key  $e$ .
- B. Creates a *RESPONSE* file (Filename: RSASigVer.rsp) containing:
  1. The information from the *REQUEST* file and
  2. An indication of whether the signature verification passed or failed for each public key/message/signature set.

Note: The IUT sends the *RESPONSE* file to the CMT laboratory for processing by the RSAVS.

The RSAVS:

- A. Compares the contents of the *RESPONSE* file with the contents of the *FAX* file.
- B. Records PASS for this test if the results for all public key/message/signature sets match; otherwise, records FAIL.

## **Appendix A References**

- [1] FIPS186-2, Digital Signature Standard (DSS), January 27, 2000.
- [2] PKCS#1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 14, 2002.
- [3] *Security Requirements for Cryptographic Modules*, FIPS Publication 140-2, National Institute of Standards and Technology, May 2001.

## **Appendix B Example of *REQUEST*, *FAX*, *RESPONSE*, and *SAMPLE* Files for RSA as approved in FIPS186-2 and specified in ANSI X9.31**

The following examples contain values that are longer than one line. These values should be on one line. For example:

```
P =
f73accd5721dad7307a70cd5c00e3d028e323781e362e17c327b239077f53cf0496b14a1fa57e
0bc18fd308fcc6c8bd2c5fcbb457bc5146cb1128f92fc9c7b3b8608e40c56c343fd0adb47c6a5d
9f55065ae42e4aab900c70fcc19cfdf9b7c19ca5118dbfc5ed4f26dd9a7dc010580c49ed2cf5
12b7239b15a1eddca82e45
```

is the character sequence 'P', <space>, '=' , <space>, 'f', '7', '3', ..., '4', '5' followed by a <newline>.

Note that these are not complete files. They are only examples of what the files would look like.

Please refer to RSAExample.zip for examples of complete files for the ANSI X9.31 RSA algorithm and for the RSASSA-PSS and RSASSA-PKCS1-v1\_5 signature schemes with appendix specified in PKCS#1 v2.1.

### **B.1 Examples of *REQUEST* Files**

#### **B.1.1 SigGenRSA.req**

```
# CAVS 3.1
# "SigGen RSA (X9.31)" information for "file1"
# Mod sizes selected: 1024 1536 2048 3072 4096
# SHA Algorithm selected: SHA-1
# Generated on Tue Apr 20 12:02:08 2004

[mod = 1024]

Msg =
921d36f40f6a4232666e59bf8bdeb00afdc60d54b190e43d732cb32dafb39caa808ef56360c1f
53d99376c750f344d5057fdd8d90ff3f097d5df5b54fa13beec41f62cf5bef7647149e1f6f7723
c51bd715f7b0d81b0e352fa30daca9d7cca26fcc560588fd8f369ff91a5f07587907fcf376e32c
55c2ef766cddbef184350e

Msg =
209cd1267bf2d930367288d5708d1ab25d79814498315708bbbd16e02eb38bb93cf886b6cdeda
e00e307a7914f008ad23fc74f4ea1affe3b33d629837044c58b78412774847a2786f94ba76edf1
57c3dce76fcefb05e24a4728af8ccbf96335b8b6d812b8361b408027ae94a49bf039b0c3d99b74
f9ea849f0464859098b998

Msg =
3a7f19005797bd8f7711fdfbefbd00213114e8e8a0bd4bdda256e71bf670916e48db6fb8022567
93665dd737da1c879f82c7dce7140e4ef9e8b81b684fa33d9c26069a30abfa7c988f3940074c25
528fb80c598bdaaaae3afdef13ac0403a2e2fd880331f91fe2e71a2da031eddf493cfa6bc79b9a
9955e9098336afbc98a687
```

```

Msg =
9c7fb045fba99d78b1da4fd391adc5e63854369c382f566d769dd4d6d0132706036b8adbcc7e45
0c4ef42fa8b45b71bf17778c5ea2d99ec1f4b3e1c5c5eb749cccd6e5ec4e1c7d03a60f0aaebcd68
416bbd855fe9f40c8c63cbff8c16d5b62939c13133ea82adc469ae75c88424fbcd52f227b82476
efa9b9c328dfb1459ccf44
.
.
.

[mod = 1536]

Msg =
82cdb2a2270caec8aecfa273532fd524a06038553e36e2150ce8bcf26e157e1be804dac6eb30f6
d0ee0420a3c1e3be5d1e6b832f82fbe9023a72db61c13ab1934986620798d7f5f615c5394323fc
b6a6ad85e30902479a32e4cdc8a3db4969bed0d9b42dd60a92814b9b55192083a7467534c0ac69
22ddec307fed8066498de6

Msg =
2b391db2e2758181cb0153ed2074dced8c272047539ab20ccc14cd61188aa1850ec2eb1075b9c8
797641cbdb50ee4df94e33047ff7c9d88b8450f16b973915a929e33144c4043541f34b2562f844
6d36bc7b76520ce476dd017a8c52c0270c8164e418d7392170e02b678a0a7e3ce6acbf479c46fa
b4370b2ddac59cf1b2f17

Msg =
a3092bb66fa68e18e8a59445e8470433a6bae719889af61938b75422fa1979153d416e266fb240
8616b0af3f1d48265f6b90f058bb30a8a7a721f1b405b568f977d8d7673e95ac054fdca9dba982
dc0ee589d24c0fa2086ab4f0a24c3c0f5f067a55cd408257b9dfbbafeae235fd0f893c2cf5bd2d
f7fdac88abceb4472e4782

Msg =
fa9bc68c16d99d978afeecb069227c51d532937470950e56130fb07bc3e6baf2e044bcd00ddaa
25bcd1329c69d44d238a8c153f243ad6d5129d6735fa14bb567f02b363b8763677e10c84a6510e
c73cacdfec495a9ab05d38f6fe908d6d78c53f9d543047ee6916627589d409181223a4f3c09afb
fb0575ede11c94643a1ac7
.

.

.

[mod = 2048]

.

.

.

[mod = 3072]

.

.

.

[mod = 4096]

```

### B.1.2 SigVerRSA.req

```

# CAVS 3.1
# "SigVer RSA (X9.31)" information for "file1"
# Mod sizes selected: 1024 1536 2048 3072 4096

```





```

S =
19f8355c89256c0b71d8fcfac972524882df6e06d5c8a867868d3a8e8faec3b3f04ee998ec77d3
cec42bdc517a67424c9242c84b41d14e3b6f28e826ede5eb6e05c544c29ae67907dd18aa8b2d31
c7fae7b615c6ba377631c6d4e43a087f7e48f4b119331cd82481b9e98967cd1f9fa72fc5e4fa3c
125324776c8a1dd5649bfc
.
.
.
[mod = 1536]
.
.
.
[mod = 2048]
.
.
.
[mod = 3072]
.
.
.
[mod = 4096]
.
.
.

```

## B.2 Examples of *FAX* Files

### B.2.1 SigGenRSA.fax

```

# CAVS 3.1
# "SigGen RSA (X9.31)" information for "file1"
# Mod sizes selected: 1024 1536 2048 3072 4096
# SHA Algorithm selected: SHA-1
# Generated on Tue Apr 20 12:02:08 2004

[mod = 1024]

Msg =
921d36f40f6a4232666e59bf8bdeb00afdc60d54b190e43d732cb32dafb39caa808ef56360c1f
53d99376c750f344d5057fd8d90ff3f097d5df5b54fa13beec41f62cf5bef7647149e1f6f7723
c51bd715f7b0d81b0e352fa30daca9d7cca26fcc560588fd8f369ff91a5f07587907fcf376e32c
55c2ef766cddbef184350e

Msg =
209cd1267bf2d930367288d5708d1ab25d79814498315708bbbd16e02eb38bb93cfe886b6cdeda
e00e307a7914f008ad23fc74f4ea1affe3b33d629837044c58b78412774847a2786f94ba76edf1
57c3dce76fcefb05e24a4728af8ccbf96335b8b6d812b8361b408027ae94a49bf039b0c3d99b74
f9ea849f0464859098b998

Msg =
3a7f19005797bd8f7711fdfbefbd00213114e8e8a0bd4bdda256e71bf670916e48db6fb8022567
93665dd737da1c879f82c7dce7140e4ef9e8b81b684fa33d9c26069a30abfa7c988f3940074c25
528fb80c598bdaaaae3afdef13ac0403a2e2fd880331f91fe2e71a2da031eddf493cfa6bc79b9a
9955e9098336afbc98a687

Msg =
9c7fb045fba99d78b1da4fd391adc5e63854369c382f566d769dd4d6d0132706036b8adbcc7e45
0c4ef42fa8b45b71bf17778c5ea2d99ec1f4b3e1c5c5eb749cc6e5ec4e1c7d03a60f0aaebcd68

```

```

416bbd855fe9f40c8c63cbff8c16d5b62939c13133ea82adc469ae75c88424fbcd52f227b82476
efa9b9c328dfb1459ccf44
.
.
.

[mod = 1536]

Msg =
82cdb2a2270caec8aecfa273532fd524a06038553e36e2150ce8bcf26e157e1be804dac6eb30f6
d0ee0420a3c1e3be5d1e6b832f82fbe9023a72db61c13ab1934986620798d7f5f615c5394323fc
b6a6ad85e30902479a32e4cdc8a3db4969bed0d9b42dd60a92814b9b55192083a7467534c0ac69
22ddec307fed8066498de6

Msg =
2b391db2e2758181cb0153ed2074dc8c272047539ab20ccc14cd61188aa1850ec2eb1075b9c8
797641cbb50ee4df94e33047ff7c9d88b8450f16b973915a929e33144c4043541f34b2562f844
6d36bc7b76520ce476dd017a8c52c0270c8164e418d7392170e02b678a0a7e3ce6acbf479c46fa
b4370b2ddac59cf1b2f17

Msg =
a3092bb66fa68e18e8a59445e8470433a6bae719889af61938b75422fa1979153d416e266fb240
8616b0af3f1d48265f6b90f058bb30a8a7a721f1b405b568f977d8d7673e95ac054fdca9dba982
dc0ee589d24c0fa2086ab4f0a24c3c0f5f067a55cd408257b9dfbbafeae235fd0f893c2cf5bd2d
f7fdac88abceb4472e4782

Msg =
fa9bc68c16d99d978afEEcb069227c51d532937470950e56130fb07bc3e6baf2e044bcdAA0ddaa
25bcd1329c69d44d238a8c153f243ad6d5129d6735fa14bb567f02b363b8763677e10c84a6510e
c73cacdfec495a9ab05d38f6fe908d6d78c53f9d543047ee6916627589d409181223a4f3c09afb
fbc575ede11c94643a1ac7
.

.

.

[mod = 2048]

.

.

.

[mod = 3072]

.

.

.

[mod = 4096]

```

## B.2.2 SigVerRSA.fax

```

# CAVS 3.1
# "SigVer RSA (X9.31)" information for "file1"
# Mod sizes selected: 1024 1536 2048 3072 4096
# SHA Algorithm selected: SHA-1
# Generated on Tue Apr 20 12:02:08 2004

[mod = 1024]

```







```
[mod = 1536]
```

```
.
```

```
.
```

```
[mod = 2048]
```

```
.
```

```
.
```

```
[mod = 3072]
```

```
.
```

```
.
```

```
[mod = 4096]
```

```
.
```

```
.
```

```
.
```

## B.3 Examples of *RESPONSE* Files

### B.3.1 SigGenRSA.rsp

```
# `SigGen931'
```

```
[mod = 1024]
```

```
n =
```

```
b82d9d45ddb8e39d1a43b0355f0037f6295ae7d1e056987eb316a60fa044f1c0356094be60b9c8  
dad3cf37576c246060dd131bf17ed036beb6f06749c7be87d1c287fec1f01e51eda76cdb68ef6f  
e0fa562d8e76cc8709243b3cdb87e4f751fa4d47b371fbb97c59c27f5450339fb2f6d93f6f699d  
37781bcb75712b80d5b6a9
```

```
e = 3
```

```
Msg =
```

```
921d36f40f6a4232666e59bf8bdeb00afdc60d54b190e43d732cb32dafb39caa808ef56360c1f  
53d99376c750f344d5057fd8d90ff3f097d5df5b54fa13beec41f62cf5bef7647149e1f6f7723  
c51bd715f7b0d81b0e352fa30daca9d7cca26fcc560588fd8f369ff91a5f07587907fcf376e32c  
55c2ef766cddbef184350e
```

```
S =
```

```
4daca70ed3da45bf8c317622bd9f9e00288a9825f9f65edba68a31c72841514092806b794d638f  
5759be57e10903d4fd55c2ccc4862237b489c33e9275a5766ce51d4772e380fab2a8f7762b1220  
a10093e51098de27ed48eeef0ccb19afc48880f5bb9a663d2b6d5fd0fc0e6c73d0d71ae91cc29e6  
baf d4ae9d108cbfe79043
```

```
Msg =
```

```
209cd1267bf2d930367288d5708d1ab25d79814498315708bbbd16e02eb38bb93cfe886b6cdeda  
e00e307a7914f008ad23fc74f4ea1affe3b33d629837044c58b78412774847a2786f94ba76edf1
```

```

57c3dce76fcefb05e24a4728af8ccbf96335b8b6d812b8361b408027ae94a49bf039b0c3d99b74
f9ea849f0464859098b998
S =
54d8111fb3d03672789b9b2b09e1f248ab944cf494a81bd6f781134d3f61e238f31ff77650a2
24d8b1f06e10519db8a53fad970ed39e13763a44d99e8cc3a906af180368a19f432a7e08f4b596
6ef18e42ce8480a62e2fb3b06debb2f352528949486d08f14871770a7390566a6ef44e79c8e474
f099d228ad2b8cc0b25262

Msg =
3a7f19005797bd8f7711fdfbefbd00213114e8e8a0bd4bdda256e71bf670916e48db6fb8022567
93665dd737da1c879f82c7dce7140e4ef9e8b81b684fa33d9c26069a30abfa7c988f3940074c25
528fb80c598bdaaaae3afdef13ac0403a2e2fd880331f91fe2e71a2da031eddf493cfa6bc79b9a
9955e9098336afbc98a687
S =
2f79cc9cd84597f1dab3998067a9a4b3792e495d6530a9ac79d8baa67981a449b2ae4dc14ec362
53a4024f5e22273a1d188063b0f33319ed7d1aeac4b43db0f36dae2b1238d3a005898433cc4c8
e9d9f372e5c643c3593bd24146e5ba0431f2012b49f15d71134c70d98becc21c1043787756edab
c31c592bbeb5fcdb15fc55

Msg =
9c7fb045fba99d78b1da4fd391adc5e63854369c382f566d769dd4d6d0132706036b8adbcc7e45
0c4ef42fa8b45b71bf17778c5ea2d99ec1f4b3e1c5c5eb749cc6e5ec4e1c7d03a60f0aaebcd68
416bbd855fe9f40c8c63cbff8c16d5b62939c13133ea82adc469ae75c88424fb52f227b82476
efa9b9c328dfb1459ccf44
S =
33269d4e0632e76f2ef75777167c18cab40692108c9fb43cd19bdbb9e3c884aa74f51021d0d98
e28f972bc132e9b89aa0216f85d9a20eb93d343fc9e4efa2345403945bf120c6b94f101ffe4e3b
009b22a079810562ced0778cc58aaec18d6762af46dbb96bedbdf71017b0070acfa0915ee90352
46041f38f84476ffc8cb93

Msg =
8aac027e833457fdb430d75f3f69f7d6918d8293818098c3dcf4f6c0df397888e456db6d4258c8
0e6eea0e85662630e24fab1fd95cf048b7f3d8d19a0b8620e2d9729ee3abbcbadc14c1302d49ed
8bc4cbd0e8d94673fedbb9ec5baad830f006af29d0488708354f0dddc4beb16412654d2266fc07
b1c11778591b560ee0f0dc
S =
24297950103b3c955a9086bf0edeb5f843336d158cd2d8dc3fc7cbe41fe73e98d056a5367f4b55
859bd07d1f5e8f74bc709b622d3ede4200acf9e24f9522e5057627c7bc0b89ea1ed18dbe17ccf4
4a07503a01e6a37c05f77e1fa5c761e839a48cda55015089fda224b5da844c60d1b0a6d46e6cc7
c4914043fbef2819ebe48e
.
.
.

[mod = 1536]

n =
9b344a18fe6073a44083e8f32cc8a3b93b4e1399839d07581b42d2174502784bbd7189fe57b8ba
295a9bbf6474681ea6c8c50910edf0c2d18b4b30762af0038d89962b29e3f7f3bf1a23376db8f8
94f08b3809827f133547c2c3c49be46aa8f906dd61315ec23d0092ffe0d8872c81feaf1aea5ed0
282bddfae9bd665a5dcab638abbb14093cc817be07f1fb5e155fec80c41e22e584ec6eb10a581
20bdb864c895b6660628f9189926c542055f055ed6865525f2c11aaaa518409da1d03a75

e = 3

Msg =
82cdb2a2270caec8aecfa273532fd524a06038553e36e2150ce8bcf26e157e1be804dac6eb30f6
d0ee0420a3c1e3be5d1e6b832f82fbe9023a72db61c13ab1934986620798d7f5f615c5394323fc

```

```

b6a6ad85e30902479a32e4cdc8a3db4969bed0d9b42dd60a92814b9b55192083a7467534c0ac69
22ddec307fed8066498de6
S =
4c17449d8d247eb0f20b6e9297363a5327c4a62d13a3f23a035123e1e2f5c1e1ce681c982ceae6
99bcc130adf5d1fa919f84e87d2fc36d9297eba3c2d4e35accadef3accdaae5daf47647e2813e9
a0b574f5e4f153433127b96d2b96cc4796f3d554e8ccf3c9717bf351eecfc3aa56caf7758497ed
db804d8945b1d3ba6aaa7dee7090f49cb43362271b154427b8dd98f46cc26279efc0ba8cdb09dc
d9f9d11195c3654d893312ba48c0ab55c75440948207d387c7b8e8fbff9f362ea2ac8fde

Msg =
2b391db2e2758181cb0153ed2074dced8c272047539ab20ccc14cd61188aa1850ec2eb1075b9c8
797641cldb50ee4df94e33047ff7c9d88b8450f16b973915a929e33144c4043541f34b2562f844
6d36bc7b76520ce476dd017a8c52c0270c8164e418d7392170e02b678a0a7e3ce6acb479c46fa
b4370b2ddac59cf1b2f17
S =
25c9044f74ed31a3aa01c83ff4ff5313dcf4c974e5c274c29023d0789a73013616114082f325cd
d5d088f10521cf1e168c2492de636a2603b95cd3afc40844ccb8879ee771c0036255bb30450888
c568854ec8681c310ee621c6460f1e1c731c62ba06665a024388fe2556ebac72b96a4a790ada73
ac48004511a4b710449830914c950f83b6f6f7145e9c2be2a747d25b784cb91756e0ea0898abff
5753c1d32456005317ea6c38f39922a1a5dbfc6bc7ad6332d8d70166aa8fc4054b1bf932

Msg =
a3092bb66fa68e18e8a59445e8470433a6bae719889af61938b75422fa1979153d416e266fb240
8616b0af3f1d48265f6b90f058bb30a8a7a721f1b405b568f977d8d7673e95ac054fdca9dba982
dc0ee589d24c0fa2086ab4f0a24c3c0f5f067a55cd408257b9dfbbafeae235fd0f893c2cf5bd2d
f7fdac88abceb4472e4782
S =
39d2f2a92f29e30c2e63b5308e2f7ff17c4aa9a1486202760bdebc90e5a8cbea583c1bf169e546
06dd7af4a43dfc0c3aaaf813dea268229e779c7714f56e802cd49fc7b6c8c55f452781f0f1c9eb
2e2307d48d59e51214ca790519b999577b2fc8bb480d36bc0aa246e53ba9a21e5af5c8beb7a3f2
20bc3c6a9769d2fe97b070bd4057135fd9c6ed6fc227d1ff5775b6a9ea443f448dfc6ba44f13ad
771978eb2e01ead0206468d73ca9a298c82b1614766cc7ea8219492c145de0878559611a

Msg =
fa9bc68c16d99d978afeecb069227c51d532937470950e56130fb07bc3e6baf2e044bcd00ddaa
25bcd1329c69d44d238a8c153f243ad6d5129d6735fa14bb567f02b363b8763677e10c84a6510e
c73cacdfec495a9ab05d38f6fe908d6d78c53f9d543047ee6916627589d409181223a4f3c09afb
fb575ede11c94643a1ac7
S =
4485cb1ec25ed8c70176e50fe70b680f7a19dae856b7723e39461423c2efc7d7e6365ac2686d25
52b5a14bb8bddd49cb70a715fab825061a0b98a45d7a8f36519939415b8c86650018b666fd2d81
ed79fb55309b22197630e868cde5141f56594f4f0fbed8b40019111899f354e480eae5c1c1fa3
f61c552830c761c97d82fbe32daa001ce6fd93d2057ac0f622d3bb35371f3ded79890f62ed7256
5c5e50fe7c8cac1d560c9eb0c9f38278998b307bf39fea08c9a482a36ccabf5265267e04
.
.
.

[mod = 2048]
.
.
.

[mod = 3072]
.
.
.
```

```
[mod = 4096]
```

```
.
```

```
.
```

```
.
```

### B.3.2 SigVerRSA.rsp

```
# `SigVer931'
```

```
[mod = 1024]
```

```
n =
de5ddda0c409f7001f206d9fd1ce3c2a4f66f773c26e9bf33a102ed58187331393c7fb5d68cb2
32bab9b7a965cfcc9cf32bc51a8df25581d9066ae537f9f9f6cdd6f27884ef5c2ec6c8ea38f082
6c9239ccdfbed393582c3b4de644ba5cf614754ed550e656a93172d81847c1fabd0b188ba0972b
dc0ad68390cbe44f308351
```

```
e = 11
```

```
Msg =
```

```
bd8ccc76ed1426ceeb4650519fb69cd908b723e8317c5bec607debc1901cbd7821895557cf362
20e1a09397a22eac3e15dad963e5f2425c9b26df1213348ebc31c563edd0913fd1fbcaa4b360b8
c7c301a1e7172debdeefaf84f63d4582fd6554079c41d968e468a38e2ab0c09c15f135d1f50cf2
b1562c045fd64c0d4e33fa
```

```
S =
```

```
3f5c8077d5687ac7886134b47a7521435b166460b164b5a9bfff924cc51303cf03863517f254eea
e19d798cb0687f11223ec476e8fe8445885589f4f84819c1d1de4093386fb649f10bdafc31114a
4f204623483a8fc26c1c60f7929b20ff415c9ceb45cbaa7f3d866fda58271a4194f8096f8fb0bf
6cedf564fa5d8b6c009216
```

```
Result = F
```

```
e = 10001
```

```
Msg =
```

```
85399028b965f57b7c2dad65c6798bbd22789ab7049c0b095de6714d6d6331a7d8f733a7fd59de
71835f441b347c99560f87ea5198bd0bda730ad2c7318482946a388a4d0c2a70aef297088ab97d
e969cf5e2175cc40289706f813dd5fa80a1f6287fce28fe0917cf25e6179fb86633e8985a8ec48
2c49873d399e589fd8b08f
```

```
S =
```

```
1e17688eb741b359a0c39d9f910a5c64b29baae78e09c2e2efb5881830269e0f569e9f5f090a1e
0d94007f6d0435b13cc1856fab3792ae7ae0af5423747929a7e26319b882bdc23dd9d2792cd3c4
7c72f0c6cf95fd187b4d14b8903f33ab90ebe6608d49830f98f180cc5ee487e883d157fa7a0296
204b170018ba4bc2b13c89
```

```
Result = F
```

```
e = 11
```

```
Msg =
```

```
dc974f267e9b0a55a2cd5a837247f3d084893be8aec6934787031260cb78fafc4da89d3621ea31
33624483f9f606c8cd0cc70e7d0de4688c4b7d5c9719d9f39fdbab1ecf5c77a1695d5d0d8432af
05a5c44b41a33b8aa84576b72c2150568d53ea384e52bcd7ed5fda7351b296b51cf35162701bc6
2de50b8f5d0b801d833ac
```

```
S =
```

```
26494feb843e28fdc8a0b56408edc58dfe9564ba80f3917179881252c7362b44ab75368153ca40
e3f5ec696e83964441313bfab77ee941e66776b37146849435ca86eba259ccb9757d7916d9300
7851e8e6978867965adec9a9f4bb2887d1e4df92950346f54c40266a74690eb46490a618bafbd9
128d0b71b895760c0230cd
```

```
Result = F
```

```
e = 11
```

```

Msg =
7b97574e79b7f1d4243ffffb01f248374287b79d3e5068a21f1b0326da5649b16576ecfc2499a4
b3a4697f798b44792dcf9b4c6b27988956bf04e8953067dc9caf716b2d84983d1ebf03f35c26a7
ba3071b01fc1ba2225db2fc094ee0a2d955ac3517ce983fee59ddd302adb96133ca908dd7a6df
3563b7d517b1394fee9a72
S =
426a8c61b0e57fde7ad1503e48f9b6ad7f0453f7fa94ff1a16d74a95db5d209bf847241f5aca07
d8ea51fce3284d1e4d6dc915d213620b65dfe2ec8462bea3de9cd9efe1c337604d51bf717b3ec3
d808982ece0db21d1f705444cda765391c67403218ab9bc9f3b809e572406db923ab2d315bfd51
66aeed76ce92aeee5d31278
Result = P

n =
de5ddda0c409f7001f206d9fd1ce3c2a4f66f773c26e9bf33a102ed58187331393c7fb5d68cb2
32bab9b7a965cfcc29cf32bc51a8df25581d9066ae537f9f9f6cdd6f27884ef5c2ec6c8ea38f082
6c9239ccdfbed393582c3b4de644ba5cf614754ed550e656a93172d81847c1fabd0b188ba0972b
dc0ad68390cbe44f308351

e = 10001
Msg =
68757b92c206317c28aee884c00e9368a9d8e698de295e2f5e1a81288a9267cfce5e7975838379
6814d2c32360bed2c1606b05b80a07a3c3755d1e203da794fc29a49b2c0beb3e1de87f430e59a3
b07c31f08dde60a4ea93183d24192e47657fc2726a1d22e78982ecdcf2ae73663582648f0474aa
a93a878ee21d7b2318e040
S =
28b3be25527504591fde5437d8654e968a36d8daa4471f8e8860db0d8fce6e782c270973143030
876687ca3544f621ba0e84549a0e702d9a4c9f06e786592994dca92093eeff73209ffd56a2eff2a
3640d331adb7bfc26b582d5a57a051c44b82dd64ffbc77119c87fa69fc73bd85136ca3e782a3f7
6a17447aec297d27b6de24
Result = P
.
.
.

n =
d68ce0dcaba96c90efc82a5d3e245339d69c0f277f02d703a8ec425f071926d10b726e32fdb740
9247763ba83bf9d77222f86bd47be737d318808a124294f3288c29b8b5c059529d8445157d9758
d46fe447c922669c018660fb1ca0a19a9e796fae23d45dc78eda5146c8fa97b0b80a14ba2dc4
cb8451589b99b357995135

e = 10001
Msg =
b7b21e2fb4ed1a7a52d55916b05e122de49825fefecd4af7e70adb308677fb7bd8ad742be0144
c12750f9047a429fdecfe22748c59c60a9bea0c9554c87a453e6394b63e00ec73f9a339f22f750
ee871067b8332a007e75e68310adb8e5a30b9a862399025acb7c1148f2770dcf28b011ef81b703
4731c4c77b5940b2d82187
S =
19f8355c89256c0b71d8fcfac972524882df6e06d5c8a867868d3a8e8faec3b3f04ee998ec77d3
cec42bdc517a67424c9242c84b41d14e3b6f28e826ede5eb6e05c544c29ae67907dd18aa8b2d31
c7fae7b615c6ba377631c6d4e43a087f7e48f4b119331cd82481b9e98967cd1f9fa72fc5e4fa3c
125324776c8a1dd5649bfc
Result = F
.
.
.

[mod = 1536]
.
.
```

```
[mod = 2048]
```

```
.
```

```
.
```

```
[mod = 3072]
```

```
.
```

```
.
```

```
[mod = 4096]
```

```
.
```

```
.
```

## B.4 Examples of *SAMPLE* Files

### B.4.1 SigGenRSA.sam

```
# CAVS 3.1
# "SigGen RSA (X9.31)" information for "file1"
# Mod sizes selected: 1024 1536 2048 3072 4096
# SHA Algorithm selected: SHA-1
# Generated on Tue Apr 20 12:02:08 2004

[mod = 1024]

n = ?

e = ?
Msg =
921d36f40f6a4232666e59bf8bdeb00afdc60d54b190e43d732cb32dafb39caa808ef56360c1f
53d99376c750f344d5057fd8d90ff3f097d5df5b54fa13beec41f62cf5bef7647149e1f6f7723
c51bd715f7b0d81b0e352fa30daca9d7cca26fcc560588fd8f369ff91a5f07587907fcf376e32c
55c2ef766cddbef184350e
S = ?

Msg =
209cd1267bf2d930367288d5708d1ab25d79814498315708bbbd16e02eb38bb93cfe886b6cdeda
e00e307a7914f008ad23fc74f4ea1affe3b33d629837044c58b78412774847a2786f94ba76edf1
57c3dce76fcefb05e24a4728af8ccbf96335b8b6d812b8361b408027ae94a49bf039b0c3d99b74
f9ea849f0464859098b998
S = ?

Msg =
3a7f19005797bd8f7711fdfbefbd00213114e8e8a0bd4bdda256e71bf670916e48db6fb8022567
93665dd737da1c879f82c7dce7140e4ef9e8b81b684fa33d9c26069a30abfa7c988f3940074c25
528fb80c598bdaaaae3afdef13ac0403a2e2fd880331f91fe2e71a2da031eddf493cfa6bc79b9a
9955e9098336afbc98a687
S = ?

Msg =
9c7fb045fba99d78b1da4fd391adc5e63854369c382f566d769dd4d6d0132706036b8adbcc7e45
0c4ef42fa8b45b71bf17778c5ea2d99ec1f4b3e1c5c5eb749cc6e5ec4e1c7d03a60f0aaebcd68
416bbd855fe9f40c8c63cbff8c16d5b62939c13133ea82adc469ae75c88424fb8d52f227b82476
efa9b9c328dfb1459ccf44
```

```

S = ?
.
.
.

[mod = 1536]

n = ?

e = ?
Msg =
82cdb2a2270caec8aecfa273532fd524a06038553e36e2150ce8bcf26e157e1be804dac6eb30f6
d0ee0420a3c1e3be5d1e6b832f82fbe9023a72db61c13ab1934986620798d7f5f615c5394323fc
b6a6ad85e30902479a32e4cdc8a3db4969bed0d9b42dd60a92814b9b55192083a7467534c0ac69
22ddec307fed8066498de6
S = ?

Msg =
2b391db2e2758181cb0153ed2074dc8c272047539ab20ccc14cd61188aa1850ec2eb1075b9c8
797641cbdb50ee4df94e33047ff7c9d88b8450f16b973915a929e33144c4043541f34b2562f844
6d36bc7b76520ce476dd017a8c52c0270c8164e418d7392170e02b678a0a7e3ce6acbf479c46fa
b4370b2ddac59cf1b2f17
S = ?

Msg =
a3092bb66fa68e18e8a59445e8470433a6bae719889af61938b75422fa1979153d416e266fb240
8616b0af3f1d48265f6b90f058bb30a8a7a721f1b405b568f977d8d7673e95ac054fdca9dba982
dc0ee589d24c0fa2086ab4f0a24c3c0f5f067a55cd408257b9dfbbafeae235fd0f893c2cf5bd2d
f7fdac88abceb4472e4782
S = ?

Msg =
fa9bc68c16d99d978afeecb069227c51d532937470950e56130fb07bc3e6baf2e044bcdaa0ddaa
25bcd1329c69d44d238a8c153f243ad6d5129d6735fa14bb567f02b363b8763677e10c84a6510e
c73cacdfec495a9ab05d38f6fe908d6d78c53f9d543047ee6916627589d409181223a4f3c09afb
fbc575ede11c94643a1ac7
S = ?

.
.
.

[mod = 2048]

.
.
.

[mod = 3072]

.
.
.

[mod = 4096]

.
.
.
```

## B.4.2 SigVerRSA.sam



